The Turing Test*

B. JACK COPELAND
The Turing Archive for the History of Computing, University of Canterbury, Private Bag 4800, Christchurch, New Zealand and Dibner Institute for the History of Science and Technology, Massachusetts Institute of Technology, Cambridge MA 02139, USA;
E-mail: bjcopeland@canterbury.ac.nz

Abstract. Turing's test has been much misunderstood. Recently unpublished material by Turing casts fresh light on his thinking and dispels a number of philosophical myths concerning the Turing test. Properly understood, the Turing test withstands objections that are popularly believed to be fatal.

1. Introduction: The State of Machine Intelligence Circa 1950

The birth of Artificial Intelligence is usually placed at approximately 1956, the year in which a program written by Newell, Simon, and Shaw – later named the Logic Theorist – successfully proved theorems from Whitehead and Russell's *Principia Mathematica*, and also the year of John McCarthy's *Dartmouth Summer Research Project on Artificial Intelligence*, the conference which gave the emerging field its name. However, this received view of the matter is not historically accurate. By 1956, computer intelligence had been actively pursued for some 10 years in Britain – under the name *machine intelligence* – and the earliest AI programs to run were written there in 1951–52. That the earliest work in the field was done in Britain is in part a reflection of the fact that the first electronic stored-program digital computers to function were built in that country (at Manchester University (the MUC, 1948) and Cambridge University (the EDSAC, 1949)). Another significant factor was the influence of Turing on the first generation of computer programmers.

Turing was thinking about machine intelligence at least as early as 1941 (D. Michie, personal communication, 1998). He is known to have circulated a typewritten paper on machine intelligence among his wartime colleagues at the Government Code and Cypher School, Bletchley Park. Now lost, this was undoubtedly the earliest paper in the field. It probably concerned machine learning and heuristic problem-solving. Both were topics that Turing discussed extensively during the war years at GC & CS, as was mechanical chess. In 1945, Turing expressed the view that a computer 'could probably be made to play very good chess' (1945: 41).

Turing's 'Proposal for Development in the Mathematics Division of an Automatic Computing Engine (ACE)' (Turing, 1945), which was written at the National Physical Laboratory, London, between October and December 1945, was the first relatively complete specification of an electronic stored-program general-purpose digital computer. The slightly earlier – and better known – 'First Draft of a Report on the EDVAC' contained little engineering detail, in particular concerning

Minds and Machines **10:** 519–539, 2000.

electronic hardware (composed by von Neumann on the basis of discussions with other members of the Moore School group at the University of Pennsylvania, the 'First Draft' was in circulation from May 1945). In contrast, Turing's 'Proposal' contained detailed circuit designs and detailed specifications of hardware units, specimen programs in machine code, and even an estimate of the cost of building the machine (£11,200). Artificial intelligence was not far from Turing's thoughts: he described himself as 'building a brain' (D. Bayley, personal communication, 1997) and he remarked in a letter to W. Ross Ashby that, in working on the ACE, he was 'more interested in the possibility of producing models of the action of the brain than in the practical applications to computing'.

In London in 1947 Turing gave what was, so far as is known, the earliest public lecture to mention computer intelligence, saying 'What we want is a machine that can learn from experience' and '[t]he possibility of letting the machine alter its own instructions provides the mechanism for this' (Turing, 1947: 123). In 1948 he wrote a report for the National Physical Laboratory entitled 'Intelligent Machinery'. This was the first manifesto of AI and in it Turing brilliantly introduced many of the concepts that were later to become central to the field, in some cases after reinvention by others. These included the *theorem-proving* approach to problem-solving, the hypothesis that 'intellectual activity consists mainly of various kinds of search' (1948: 23), the concept of a *genetic algorithm*, and, anticipating connectionism, the proposal that networks of artificial neurons be 'trained' (Turing's word) to perform specific tasks (see further Copeland and Proudfoot, 1996, 1999a). The major part of the report consists of an exquisite discussion of machine learning. Turing describes some experiments concerning the modification of an initially 'unorganised machine', by a process akin to teaching by reward and punishment (1948: 17–21). He subsequently referred to this unorganised machine as a 'child-machine' (1950a: 457), saying that he had 'succeeded in teaching it a few things, but the teaching method was too unorthodox for the experiment to be considered really successful' (ibid.: 457). The report ends with a description of a restricted form of what Turing was later to call the 'imitation game':

> It is possible to do a little experiment ... even at the present stage of knowledge. It is not difficult to devise a paper machine which will play a not very bad game of chess. Now get three men as subjects for the experiment, A, B, and C. A and C are to be rather poor chess players. B is the operator who works the paper machine. (In order that he should be able to work it fairly fast it is advisable that he be both mathematician and chess player.) Two rooms are used with some arrangement for communicating moves, and a game is played between C and either A or the paper machine. C may find it quite difficult to tell which he is playing. (This is a rather idealized form of an experiment I have actually done.) (1948: 23)

In 1951, Turing gave a lecture on machine intelligence on British radio (Turing, 1951a), part of which is reproduced in Section 6 below. One of Turing's listeners was Christopher Strachey. To Strachey belongs the honor of having written the

earliest AI program, a checkers (or draughts) player which first ran successfully in Turing's Computing Machine Laboratory at Manchester University. At the time Strachey was a schoolmaster at Harrow; he later became Director of the Programming Research Group at Oxford University, where with Dana Scott he did the work on the semantics of programming languages for which he is best known. Strachey initially coded his checkers program in May 1951 for the Pilot Model of the ACE at the National Physical Laboratory. This version of the program did not run successfully; Strachey's efforts were defeated first by coding errors and subsequently by a hardware change that rendered his program obsolete. Strachey transferred his allegiance to the Manchester laboratory, which in February 1952 took delivery of a Ferranti Mark I computer; built by the Manchester firm of Ferranti, in close collaboration with the Computing Machine Laboratory, the Mark I was the world's first commercially available electronic stored-program computer. This machine had more memory than the Pilot ACE and was better suited to Strachey's purposes. With Turing's encouragement, and using Turing's recently completed *Programmers' Handbook* for the Ferranti machine (Turing, 1950b), Strachey finally got his program working. By the summer of 1952 the program could play a complete game of checkers at a reasonable speed. Strachey's program used simple heuristics and looked ahead 3–4 turns of play. The state of the board was represented on the face of a cathode ray tube – one of the earliest uses of computer graphics.

In 1952, Strachey described his checkers program at a computing conference in North America. Arthur Samuel of IBM took over the essentials of Strachey's program and coded a checkers player for the IBM 701. This was the first AI program to function in the U.S. In 1955 Samuel added learning to the program.

Strachey himself was thinking about machine learning at the time of writing his checkers player, devising a simple rote-learning scheme that he envisaged being implemented in a NIM-playing program. Strachey wrote at length concerning machine learning in a letter to Turing that he composed on the evening of Turing's radio broadcast. The following is a short extract from this letter:

> I have just been listening to your talk on the Third Programme. Most stimulating ... [i]n paricular your remark ... that the programme for making a machine think would probably have great similarities with the process of teaching; this seems to me absolutely fundamental. ... I am convinced that the crux of the problem of learning is recognizing relationships and being able to use them. ... There are, I think, three main stages in learning from a teacher. The first is the exhibition of a few special cases of the rule to be learned. The second is the process of generalisation – i.e. the underlining of the important features that these cases have in common. The third is that of verifying the rule in further special cases and asking questions about it. I have omitted any mention of 'understanding' the rule, because this is not appropriate at the moment to the action of a machine. I think, as a matter of fact, that the process of understanding a rule is connected with finding relationships between it and other rules – i.e. second (or higher) order relations between relations and this might well become

> important for a machine later. ... I think it might well be possible to programme the Manchester machine to do all of these stages, though how much it would be able to learn in this way before the storage became inadequate remains to be seen. ... (Strachey to Turing, 15 May 1951)

However, Strachey missed the opportunity to be the first to achieve a functioning program that incorporated learning. The earliest such programs were written by Anthony Oettinger, at the University of Cambridge Mathematical Laboratory, home of the EDSAC. Oettinger was considerably influenced by Turing's views on machine learning. Oettinger's 'response-learning programme,' details of which were published in 1952, could be taught to respond appropriately to given stimuli, by means of expressions of 'approval' or 'disapproval' by the teacher. As training proceeded, errors became less frequent, and the learned response would be initiated by a progressively weaker stimulus. Oettinger described the response-learning program as 'operating at a level roughly corresponding to that of conditioned reflexes,' remarking that the 'behaviour pattern of the response-learning ... machine is sufficiently complex to provide a difficult task for an observer required to discover the mechanism by which the behaviour of the ... machine is determined.'

Oettinger's 'shopping machine', also reported in 1952, incorporated rote-learning. Adopting Turing's terminology, Oettinger described this program as a 'child machine.' The shopping machine's simulated world was a mall of eight shops. When sent out to purchase an item, Shopper would if necessary search for it, visiting shops at random until the item was found. While searching, Shopper would memorise a few of the items stocked in each shop that it visited. Next time Shopper was sent out for the same item, or for some other item that it had already located, it would go to the right shop straight away.

Oettinger was the first programmer to claim a program capable of passing a restricted Turing test. Shopper, he said, could successfully play a version of Turing's imitation game in which

> the questions are restricted to ... the form 'In what shop may article j be found?'

2. Did Turing Offer a Definition of 'Thinking' (or 'Intelligence')?

In his classic paper on the Turing test, Moor (1976) wrote:

> [T]he proponents and critics of the imitation game have misunderstood its significance. The real value of the imitation game lies not in treating it as the basis for an operational definition but in considering it as a potential source of good inductive evidence for the hypothesis that machines think. (1976: 249; see also Moor, 1987)

Twenty-five years later, the lesson has still not been learned that there is no *definition* to be found in Turing's paper of 1950. Commentator after commentator states that Turing's intention was to offer a definition of 'thinking' or 'intelligence'. French, a prominent critic of the Turing test, asserts:

> The Turing Test [was] originally proposed as a simple operational definition of intelligence. (French, 2000: 115)

Perhaps the publication of the little-known remarks by Turing that appear in Section 3, below, will scupper this misunderstanding once and for all.

Unfortunately, Turing's biographer, Andrew Hodges, endorsed the claim that Turing's intention was to provide a definition. Hodges' statement that Turing

> introduced ... an operational definition of 'thinking' or 'intelligence' ... by means of a sexual guessing game (1992: 415)

has no doubt misled many.

It is difficult to reconcile the suggestion that Turing intended to propose a definition of 'thinking' or 'intelligence' with his explicit statement in the 1950 paper that machines may

> carry out something which ought to be described as thinking but which is very different from what a man does. (1950a: 435)

Nevertheless, some commentators who notice this remark persist in the claim that Turing put forward a definition. Block writes as follows:

> An especially influential behaviorist definition of intelligence was put forward by Turing [1950a]. ... Turing's version of behaviorism formulates the issue of whether machines could think or be intelligent in terms of whether they could pass the following test ... The computer is intelligent *if and only if* the judge cannot tell the difference between the computer and the person. (Osherson and Lasnik, 1990: 248, my italics)

Then, in a footnote, in which he mentions the statement by Turing just quoted, Block remarks that Turing

> jettisoned the claim that being able to pass the Turing test is a necessary condition of intelligence, weakening his claim to: passing the Turing test is a sufficient condition for intelligence. (ibid.: 249–250)

However, Turing never claimed in the first place that the ability to pass the Turing test is a necessary condition for intelligence; it is entirely misleading to describe him as *jettisoning* this necessary condition and *weakening* his claim. (Sadly, these misleading claims appear in a widely-read textbook.)

In fact, as we shall see, Turing explicitly denies that he is proposing a definition. The material by Turing presented in Section 3 may also assist in the eradication of another myth, namely that Turing did not intend to propose a *test*. Whitby complains about

> the change from the label 'imitation game' to 'Turing test' by commentators,

and he says that the

> suggestion that the [imitation game] might be some sort of test involves an important extension of Turing's claims. (1996: 54)

Similarly, Narayanan writes:

> Turing did not originally intend his imitation game to be a test as such. (1996: 66)

Turing did indeed intend his game to be a test.

3. Turing's 1952 Presentation of the Imitation Game

The following is an extract from the typewritten script of a BBC radio broadcast entitled 'Can Automatic Calculating Machines Be Said To Think', recorded in January 1952.[1] In response to the introductory remarks

> We're here today to discuss whether calculating machines can be said to think in any proper sense of the word. ... Turing, ... [h]ave you a mechanical definition?,

Turing replies:

> I don't want to give a definition of thinking, but if I had to I should probably be unable to say anything more about it than that it was a sort of buzzing that went on inside my head. But I don't really see that we need to agree on a definition at all. The important thing is to try to draw a line between the properties of a brain, or of a man, that we want to discuss, and those that we don't. To take an extreme case, we are not interested in the fact that the brain has the consistency of cold porridge. We don't want to say 'This machine's quite hard, so it isn't a brain, and so it can't think.' I would like to suggest a particular kind of *test* that one might apply to a machine. You might call it a test to see whether the machine thinks, but it would be better to avoid begging the question, and say that the machines that pass are (let's say) 'Grade A' machines. The idea of the test is that the machine has to try and pretend to be a man, by answering questions put to it, and it will only pass if the pretence is reasonably convincing. A considerable proportion of a jury, who should not be expert about machines, must be taken in by the pretence. They aren't allowed to see the machine itself – that would make it too easy. So the machine is kept in a far away room and the jury are allowed to ask it questions, which are transmitted through to it: it sends back a typewritten answer. ... [The questions can concern] anything. And the questions don't really have to be questions, any more than questions in a law court are really questions. You know the sort of thing. 'I put it to you that you are only pretending to be a man' would be quite in order. Likewise the machine would be permitted all sorts of tricks so as to appear more man-like, such as waiting a bit before giving the answer, or making spelling mistakes, but it can't make smudges on the paper, any more than one can send smudges by telegraph. We bad better suppose that each jury has to judge quite a number of times, and that sometimes they really are dealing with a man and not a machine. That will prevent them saying 'It must be a machine' every time without proper consideration.

> Well, that's my test. Of course I am not saying at present either that machines really could pass the test, or that they couldn't. My suggestion is just that this is the question we should discuss. It's not the same as 'Do machines think,' but it seems near enough for our present purpose, and raises much the same difficulties.

The 1952 exposition significantly modifies the arrangements described by Turing in 1950. According to the earlier formulation, the Turing test is a three-party game involving the parallel interrogation by a human of a computer and a human foil. According to the 1952 formulation, members of a jury question a series of contestants one by one, some of the contestants being machines and some humans. (The arrangements adopted in the Loebner series of Turing tests, sometimes condemned as improper, in fact conform to Turing's 1952 formulation of his test.) The later formulation abandons the condition, implicit in the earlier, that each interrogator shall know that one of each pair of contestants is a human and one a machine. It appears that the 1950 formulation is superior, since the test as formulated in 1952 is open to a biassing effect, which disfavours the machine. Results of the Loebner series of competitions reveal a strong propensity among jurors to classify human respondents as machines. In the Loebner competition held at Dartmouth College in January 2000, human respondents were mistaken for computers on 10 occasions, a computer for a human on none. The same effect was present in similarly structured tests performed with Colby's program Parry (Heiser, Colby et al., 1980). In a total of 10 interviews there were five misidentifications; in four of these a human respondent was mistaken for a computer. Presumably this phenomenon is the result of a determination on the part of the jurors not to be fooled by a program. This lengthening of the odds against the machine cannot occur in the three-player form of the test.

It is interesting that, in the 1952 formulation, Turing specifically excludes computer scientists from the jury. This is certainly in the same spirit as his remark in 1948, quoted above, that jurors in the chess-player version of the test should be 'rather poor' at chess. It is often pointed out (for instance by Douglas Lenat in a paper presented at Turing 2000) that certain characteristic weaknesses in human reasoning – for example, a willingness in certain circumstances to assign a lower probability to a conjunct than to the conjunction, or the tendency to fail to take notice of certain disconfirming instances of conditional statements (weaknesses which are easily detectable by Wason tests and the like) – could be used to unmask any computer not specifically programmed to reproduce these human foibles. However, it is likely that had Turing been writing after the relevant empirical discoveries by Wason, Johnson-Laird, Taversky, Kahnemann, and others, that he would have excluded from the jury not only those who are 'expert about machines' but also those who are expert about the human mind.

In the 1950 formulation, Turing introduces his test by first describing an imitation game involving an interrogator and two subjects, one male (A) and one female (B). The interrogator must determine, by question and answer, which of A and B

is the man (A's object in the game being to try to cause the interrogator to make the wrong identification). Turing then asks 'What will happen when a machine takes the part of A in this game?' (1950a: 434). However, later in the paper he describes matters a little differently, saying that the part of A is taken by a machine and 'the part of B ... by a man' (1950a: 442). Sterrett (this volume) distinguishes between what she calls the 'original imitation game test', or 'OIG test', and what she calls the 'standard Turing test', which she says is described in the second of these passages. In the 'OIG test', the computer attempts to impersonate a woman, and its degree of success is compared with a male player's degree of success at impersonating a woman. If one sees the tests that Sterrett distinguishes as different, one may wonder which Turing intended to advocate (perhaps both). Traiger (this volume) argues that Turing was advocating the OIG test. However, in the 1952 formulation, Turing says simply that '[t]he idea of the test is that the machine has to try and pretend to be a man ... and it will pass only if the pretence is reasonably convincing.' Indeed, in the little-known material concerning the test contained in Section 6 below, which dates from 1951, Turing presents matters in a starkly ungendered form: the point of the test is to determine whether or not a computer can 'imitate the brain'. It seems unlikely, therefore, that Turing's intention in 1950 was to endorse only the female-impersonator form of the test, or that he saw himself as describing different tests in the passages in question.

There is another minor tension in the 1950 paper. Turing says both that

> [t]he original question, 'Can machines think?' I believe to be too meaningless to deserve discussion (1950a: 442)

and that

> [w]e cannot altogether abandon the original form of the problem [viz. 'Can machines think?'], for opinions will differ as to the appropriateness of the substitution and we must at least listen to what has to be said in this connexion (ibid.).

Turing was no doubt overstating his case in the first of these quotations. In the 1952 material, Turing's attitude to the question 'Can machines think?' is much milder; and likewise in the material appearing in Section 6, where Turing makes liberal use of such phrases as 'programm[ing] a machine ... to think' and 'the attempt to make a thinking machine'.

4. The 1950 and 1952 Predictions

In 1950 Turing predicted that

> in about fifty years' time it will be possible to programme computers ... to make them play the imitation game so well that an average interrogator will not have more than 70 per cent chance of making the right identification after five minutes of questioning. (1950a: 442)

It has been said that the outcome of the 2000 Loebner Turing test shows Turing's prediction to have been in error. But far from it. 'About fifty years' does not mean 'exactly fifty years'. Moreover, Turing's prediction is a fairly modest one: 3 out of 10 average interrogators are to make a wrong identification on the basis of no more than five minutes questioning. (Turing's prediction is sometimes mis-reported (e.g. by Whitby 1996: 61) as being the claim that, by the end of the 20th century, computers would succeed in deceiving the interrogator 70 percent of the time.) The qualification 'average' presumably indicates that the interrogators are not to be computer scientists, psychologists, or others whose knowledge or training is likely to render them more skilled at the task than an average member of the population.

In 1952, in an exchange with another contributor to the same BBC broadcast, the Cambridge mathematician Max Newman,[2] Turing offered a prediction that is interestingly different from the above:

Newman: I should like to be there when your match between a man and a machine takes place, and perhaps to try my hand at making up some of the questions. But that will be a long time from now, if the machine is to stand any chance with no questions barred?

Turing: Oh yes, at least 100 years, I should say.

This prediction perhaps concerns success of a more substantial nature than that envisaged in the much-quoted 1950 prediction. Possibly Turing would have endorsed the result of precising the 1952 prediction in the way proposed in the 1950 paper:

> We now ask the question, "What will happen when a machine takes the part of A in this [man-imitates-woman] game?" Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman? (1950a: 434)

Reformulating the 1952 prediction in this fashion produces: It will be at least 100 years (2052) before a computer is able to play the imitation game sufficiently well so that judges will decide wrongly as often in man-imitates-woman imitation games as in machine-imitates-man (or human) imitation games, in each case no questions being barred.

5. De Cordemoy's Anticipation of the Turing Test in 1668

Descartes famously declared that it is

> not conceivable that ... a machine should produce different arrangements of words so as to give an appropriately meaningful answer to whatever is said in its presence, as the dullest of men can do.[3]

The Cartesian Géraud de Cordemoy wrote at some length on the matter of distinguishing between that which thinks and that which does not. The following extracts are from his *A Philosophicall Discourse Concerning Speech* of 1668.

> To speak is not to repeat the same words, which have struck the ear, but to utter others to their purpose and suitable to them. ... [N]one of the bodies that make echoes do think, though I hear them repeat my words ... I should by the same reason judge that parrets do not think neither. ... But not to examine any further, how it is with parrets, and so many other bodies, whose figure is very different from mine, I shall continue the inquiry ... [Concerning those] who resemble me so perfectly *without* ... I think I may ... establish for a Principle, that ... if I finde by all the experiments I am capable to make, that they use speech as I do, ... I have infallible reason to believe that they have a soul as I. (1668: 13–14)

I dub the following *de Cordemoy's Principle*: If all the experiments that we are capable of making show that x uses speech as we do, then x has a soul (i.e. thinks).

De Cordemoy himself assumed, not surprisingly, that a machine would always be easily unmasked.

> I do very well conceive, that a meer Engin might utter some words, yet I understand at the same time, that if the organs, which should distribute the wind, or open the pipes, whence those voices should issue, had a certain settled order among them, they could never change it, so that when the first voice were heard, those that were wont to follow it, would needs be heard also, provided the wind were not wanting to the Engin; whereas the words which I hear uttered by bodies made like mine have almost never the same sequel. (ibid.: 6)

Of course, even this Engin might on some occasions be mistaken for a thinking thing by 17th century folk untutored in the ways of machinery. However, experiments involving more discriminating judges (or even further experiments involving the same judges) would easily reveal that in fact the Engin does not use speech as we do. By insisting that x perform satisfactorily in all experiments, de Cordemoy's principle allows for the fact that the results of some may be misleading. In modern terms, a machine that happens to pass one Turing test, or even a series of them, might be shown by subsequent tests to be a relatively poor player of the imitation game.

In a trenchant critique of the Turing test, Shieber imputes to Turing the view that

> any agent that can be mistaken by virtue of its conversational behavior [for] a human must be intelligent. (1994: 70)

This view, supposedly embodied in the Turing test, has been the target of much criticism. The fact that a program has been mistaken for a human by a particular set of interrogators may tell only of the gullibility of those interrogators; or, indeed, the program may by good luck have given a thoroughly atypical performance, in the fashion of a first-season football star whose performance subsequently regresses to the mean. Therefore, it is argued, a positive outcome of the Turing test cannot be sufficient for the claim that the successful machine thinks. However, there is no reason to believe that Turing is any more vulnerable to this objection than is de Cordemoy. Turing's position as described by Turing is entirely consistent with the

Cordemoy-like view that the result of any given experiment is defeasible and may be disregarded in the light of other experiments.

6. Turing's Remarks on the Foundation of the Turing Test

The following is an extract from Turing's lecture 'Can Digital Computers Think?', which was broadcast on BBC radio in May 1951.[4] Turing's was the second in a series of lectures with the general title 'Automatic Calculating Machines.' Other speakers in the series included Newman, Hartree, Wilkes, and F.C. Williams. (The extract is from Turing's own typescript.)

> I believe that [digital computers] could be used in such a manner that they could appropriately be described as brains. ... This ... statement needs some explanation. ... In order to arrange for our computer to imitate a given machine it is only necessary to programme the computer to calculate what the machine in question would do under given circumstances ... If now some particular machine can be described as a brain we have only to programme our digital computer to imitate it and it will also be a brain. If it is accepted that real brains, as found in animals, and in particular in men, are a sort of machine it will follow that our digital computer suitably programmed will behave like a brain. This argument involves [the assumption] which can quite reasonably be challenged ... that this machine should be of the sort whose behaviour is in principle predictable by calculation. ...
>
> [A]lthough [digital computers] might be programmed to behave like brains, we do not at present know how this should be done. ... [A]s to whether we will or will not eventually succeed in finding such a programme ... I, personally, am inclined to believe that such a programme will be found. I think it is probable for instance that at the end of the century it will be possible to programme a machine to answer questions in such a way that it will be extremely difficult to guess whether the answers are being given by a man or by the machine. I am imagining something like a viva-voce examination, but with the questions and answers all typewritten in order that we need not consider such irrelevant matters as the faithfulness with which the human voice can be imitated. ...
>
> [O]ur main problem [is] how to programme a machine to imitate the brain, or as we might say more briefly, if less accurately, to think. ... The fact is that we know very little about [how to do this]. ... The whole thinking process is still rather mysterious to us, but I believe that the attempt to make a thinking machine will help us greatly in finding out how we think ourselves.

I dub the following *Turing's Principle*: A machine that by means of calculation imitates – or, better, 'emulates,' for Turing is concerned with faithful imitation[5] – the intellectual behaviour of a human brain can itself appropriately be described as a brain, or as thinking. (Only the intellectual behaviour of the brain need be

emulated: ‘we are not interested in the fact that the brain has the consistency of cold porridge’.)

It is often claimed that Turing was insufficiently specific in his description of his test. What are the specifications of a *definitive* test? How long? How many judges? What number of correct identifications is to be tolerated? However, these demands appear to miss the point. Whether a given machine is able to emulate the brain is not the sort of matter that can be settled conclusively by a test of brief duration. A machine emulates the brain if it plays the imitation game successfully come what may, with no field of human endeavour barred, and for any length of time commensurate with the human lifespan. Consider two time-unlimited imitation games, a man-woman game and a machine-human game, each employing the same diversity of judges that one might encounter, say, on the New York subway. If, in the long run, the machine is identified correctly no more often than is the man in the man-woman game, then the machine is emulating the brain. Any test short enough to be practicable is but a sampling of this ongoing situation. After some amount of sampling, we may become convinced that, in the long run, the machine will play as well as the man, but only because we believe that our samples of the machine’s performance are representative, and we may always change our opinion on the basis of further rounds of the game.

Turing’s position appears to consist of essentially three components: (1) Turing’s Principle; (2) the claim that the method of question and answer provides a suitable means for determining whether or not a machine is able to emulate human intellectual behaviour; and further (3) that the imitation game, in its specific provisions, is suitable for this purpose. How might Turing have defended these individual propositions?

Turing speaks elsewhere of the need to give ‘fair play’ to machines (e.g. Turing, 1947: 123). It is perhaps in some such terms that Turing would seek to defend his claim that whatever emulates the brain can appropriately be described as thinking. As for the second claim, Turing offers an explicit justification: the ‘question and answer method’

> draw[s] a fairly sharp line between the physical and the intellectual capacities of a man

and

> seems to be suitable for introducing almost any one of the fields of human endeavour that we wish to include. (1950a: 434–435)

No explicit justification of the third claim is given. Some doubts about the truth of the claim – for example, that the outcome of the test is a function of the gullibility of the jury – can be allayed as above, by pointing out that (there is no reason to think Turing would have denied that) the upshot of any given round of play of the imitation game may be disregarded in the light of further rounds.

Turing’s phraseology in the 1950 and 1952 presentations sometimes calls to mind his proposal (Turing, 1936) that such questions as “Is there an effective method for solving such-and-such a mathematical problem?” be replaced by the

clear and unambiguous “Is there a Turing machine for solving the problem in question?” (a proposal now known as the Church-Turing thesis):

> I shall replace the question [‘Can machines think?’] by another, which is closely related to it and is expressed in relatively unambiguous words. (1950a: 433)

Some pages later Turing describes this proposal as a *tentative suggestion*:

> It was suggested tentatively that the question ‘Can machines think?’ should be replaced by ‘Are there imaginable digital computers which would do well in the imitation game?’ (1950a: 442)

What in 1936 was also something of a tentative suggestion had found widespread support by the time of Turing’s paper of 1948 (and since that date, of course, even more logico-mathematical evidence has amassed in favour of the suggestion):[6]

> It is found in practice that LCMs [‘logical computing machines’ – Turing’s expression for (what Church called) Turing machines] can do anything that could be described as ‘rule of thumb’ or ‘purely mechanical.’ This is sufficiently well established that it is now agreed amongst logicians that ‘calculable by means of an LCM’ is the correct accurate rendering of such phrases. (Turing, 1948: 7)

Certainly no such consensus has yet arrived in the case of Turing’s later suggestion. But nor, despite 50 years of vigorous and sometimes highly ingenious criticism, has there been a successful attempt to refute it.

One obvious objection to Turing’s proposal is that it involves an anthropocentric bias. As Turing himself puts the objection:

> The game may perhaps be criticised on the ground that the odds are weighted too heavily against the machine. ... May not machines carry out something which ought to be described as thinking but which is very different from what a man does? (1950a: 435)

It is important to keep in mind that Turing’s proposal concerns the – as it were, existentially quantified – question ‘Can machines think?’. Moreover, Turing advanced the proposal in the belief that computers will play the imitation game successfully. There might indeed be questions of the form ‘Can machine *M* think?’ which could not be settled by the imitation game. Turing’s opinion is presumably that any question of this form that cannot be so settled is insufficiently clear in meaning to admit of an unambiguous answer: only where the replacement by ‘Can machine *M* do well in the imitation game?’ is appropriate does the question ‘deserve discussion’. At best, the objection shows only that if no successful player of the imitation game were to emerge, the question ‘Can machines think?’ might remain unsettled. Turing cheerfully sets aside this possibility:

> [The] objection is a very strong one, but at least we can say that if, nevertheless, a machine can be constructed to play the imitation game satisfactorily, we need not be troubled by this objection. (ibid.)

7. Attempts to Discredit the Turing Test

These have, of course, been many and varied. I select three that are especially prominent. (Shieber, for example, singles these out, describing them as 'strong' (1994: 74).)

7.1 BLOCKHEADS

A 'blockhead' (after Block, 1981) is a hypothetical program able to play the imitation game successfully, for any fixed length of time, by virtue of incorporating a large, but finite, 'lookup' table containing all the exchanges with the interrogator that could occur during the length of time in question. Such a program emulates the intellectual behaviour of the brain but (it is assumed) does not think, in contradiction to Turing's Principle. I will call this the 'lookup table objection'. Although usually credited to Block, this objection to the Turing test has occurred to a number of writers. The earliest known presentation of the objection was by Shannon and McCarthy in 1956:

> The problem of giving a precise definition to the concept of 'thinking' and of deciding whether or not a given machine is capable of thinking has aroused a great deal of heated discussion. One interesting definition has been proposed by A.M. Turing: a machine is termed capable of thinking if it can, under certain prescribed conditions, imitate a human being by answering questions sufficiently well to deceive a human questioner for a reasonable period of time. A definition of this type has the advantages of being operational, or, in the psychologists' term, behavioristic. ... A disadvantage of the Turing definition of thinking is that it is possible, in principle, to design a machine with a complete set of arbitrarily chosen responses to all possible input stimuli ... Such a machine, in a sense, for any given input situation (including past history) merely looks up in a 'dictionary' the appropriate response. With a suitable dictionary such a machine would surely satisfy Turing's definition but does not reflect our usual intuitive concept of thinking. This suggests that a more fundamental definition must involve something relating to the manner in which the machine arrives at its responses – something which corresponds to differentiating between a person who solves a problem by thinking it out and one who has previously memorized the answer. (1956: v–vi)

The formal point on which the objection rests – essentially that if the timespan of the imitation game is (finite and) bounded, and if the rate at which characters can be typed is likewise bounded, then all possible interchanges can be inscribed on a finite segment of the tape of a Turing machine – would have been obvious to Turing. In the 1950 paper, he points out that the behaviour of any discrete system with a finite number of configurations can be represented by a finite lookup table, and that a computer can mimic the system if supplied with the table:

> discrete state machines ... can be described by such tables provided they have only a finite number of possible states. ... Given the table corresponding to a discrete state machine ... [and provided the calculation] could be carried out sufficiently quickly the digital computer could mimic the behaviour of [the] discrete state machine. The imitation game could then be played with the machine in question (as B) and the mimicking digital computer (as A) and the interrogator would be unable to distinguish them. Of course the digital computer must have an adequate storage capacity as well as working sufficiently fast. (1950a: 440–441)

What Turing might have said in response to Shannon and McCarthy may perhaps be inferred from his caveats concerning storage capacity and speed, especially if those remarks are taken in conjunction with the following (from the broadcast ‘Can Automatic Calculating Machines Be Said to Think?’):

Newman: It is all very well to say that a machine could ... be made to do this or that, but, to take only one practical point, what about the time it would take to do it? It would only take hour or two to make up a routine to make our Manchester machine analyse all possible variations of the game of chess right out, and find the best move that way – *if* you didn’t mind its taking thousands of millions of years to run through the routine. Solving a problem on the machine doesn’t mean finding a way to do it between now and eternity, but within a reasonable time. ...

Turing: To my mind this time factor is the one question which will involve all the real technical difficulty. If one didn’t know already that these things can be done by brains within a reasonable time one might think it hopeless to try with a machine. The fact that a brain *can* do it seems to suggest that the difficulties may not really be so bad as they now seem.

The answer to Shannon, McCarthy, Block et al, that these remarks suggest is this: firstly, the proposed recipe for building a brain-emulator cannot work, given practical limitations on storage capacity; and secondly, even if this point is set aside and we suppose that such a machine were actually to be constructed, it would *not* emulate the brain, since what the brain can do in minutes would take this machine ‘thousands of millions of years.’

If Turing had been proposing a definition of ‘thinking’ – a logically necessary and sufficient condition – or even merely a logically sufficient condition, then the lookup table objection would indeed count against the proposal, since the objection establishes that ‘if x plays the imitation game satisfactorily, then x thinks’ is false in some possible world very different from the actual world. However, there is no reason to believe that Turing was claiming anything more than that his principle is actually true. The other-worldly possibility of a lookup-table machine that is fast enough to emulate the brain has no tendency at all to show that Turing’s principle is *actually* false. (Likewise, it is no challenge to the actual truth of the Church-Turing thesis that a human rote-worker who occupies a possible world in which the human memory is unlimited can be in the process of writing down a number that is not computable by any Turing machine (see Turing, 1936: 231, 249–252).)

7.2. THE CHINESE ROOM

As is well-known, Searle (1980) considers the case of a human clerk – call him or her Clerk – who ‘handworks’ a computer program that is capable of passing the Turing test in Chinese. Clerk is a monolingual English speaker. The program is presented to Clerk in English in the form of a set of rule-books. Clerk works in a room concealed from the interrogator’s view; he and the interrogator communicate by passing sheets of paper through a slot. To the interrogator, the verbal behaviour of the Room – the system that includes the rule-books, Clerk, the erasable paper memory, Clerk’s pencils and rubbers, the input and output provisions, and any clock, random number generator, or other equipment that Clerk may need in order to execute the program in question – is by hypothesis indistinguishable from that of a native Chinese speaker.

Here is Searle’s argument:

> [Clerk] do[es] not understand a word of the Chinese ... [Clerk] ha[s] inputs and outputs that are indistinguishable from the native Chinese speaker, and [Clerk] can have any formal program you like, but [Clerk] still understand[s] nothing. [A] computer for the same reasons understands nothing ... [W]hatever purely formal principles you put into the computer will not be sufficient for understanding, since a human will be able to follow the formal principles without understanding ... (1980: 418)

> [Clerk] can pass the Turing test; [Clerk] can fool native Chinese speakers. ... The example shows that there could be two ‘systems’, both of which pass the Turing test, but only one of which understands ... (ibid.: 419)

The flaw is a simple one: Searle’s argument is *not logically valid*. The proposition that the formal operations carried out by Clerk do not enable Clerk to understand the Chinese inputs and outputs by no means entails the quite different proposition that the formal operations carried out by Clerk do not enable the Room to understand the Chinese inputs and outputs. One might as well claim that the statement ‘The organisation of which Clerk is a part has no taxable assets in Japan’ follows logically from the statement ‘Clerk has no taxable assets in Japan’. Searle’s example does not show, therefore, that there could be a system which passes the Turing test but which does not understand (think).

It is important to distinguish this, *the logical reply* to the Chinese room argument (Copeland, 2001, 1993), from what Searle calls the *systems reply*. The systems reply is the following claim:

> While it is true that the individual person who is locked in the room does not understand the story, the fact is that he is merely part of a whole system and the system does understand the story. (ibid.: 419)

As Searle correctly points out, the systems reply is worthless, since it ‘simply begs the question by insisting without argument that the system must understand Chinese’ (ibid.). The logical reply, on the other hand, is a point about entailment.

The logical reply involves no claim about the truth – or falsity – of the statement that the Room can understand Chinese.

7.3. French: associative priming and rating games

As already mentioned, French (1990, 2000) is among those who mistakenly interpret Turing as offering a definition of intelligence. French, moreover, takes Turing to be offering a definition of 'intelligence in general.' French argues against Turing's supposed 'operational definition of intelligence,' objecting that

> the Test provides a guarantee not of intelligence but of culturally-oriented *human* intelligence ... [T]he Turing Test [is] a test for *human* intelligence, not intelligence in general. (1990: 12)[7]

Here French is simply missing the point. Far from Turing's offering the imitation game as a test for 'intelligence in general,' Turing did intend it precisely as a means for determining whether or not a given machine emulates the human brain.

More interestingly, French also 'take[s] issue with Turing's ... claim ... that in the not-too-distant future it [will] in fact be possible actually to build ... a machine' that plays the imitation game successfully. French believes that the imitation game's 'very strength becomes a weakness': since the 'game ... provides a very powerful means of probing humanlike cognition,' only 'a machine capable of experiencing the world in a manner indistinguishable from a human being' is likely to enjoy success in the game (ibid.: 15, 25). French illustrates his claim with ingenious examples.

7.3.1. Associative Priming

Empirical studies show that in word/non-word recognition tasks, subjects take less time to determine that an item is a word if presentation of the item is preceded by presentation of an associated word (e.g. prior presentation of 'bread' facilitates recognition that 'butter' is a word). French writes:

> The Turing Test interrogator makes use of this phenomenon as follows. The day before the Test, she selects a set of words (and non-words), runs the lexical decision task on the interviewees and records average recognition times. She then comes to the Test armed with the results ... [and] identifies as the human being the candidate whose results more closely resemble the average results produced by her sample population of interviewees. The machine would invariably fail this type of test because there is no a priori way of determining associative strengths ... Virtually the only way a machine could determine, even on average, all of the associative strengths between human concepts is to have experienced the world as the human candidate and the interviewers had. (1990: 17)

French's proposal is illegitimate. The specifications of the Turing test are clear: the interrogator is allowed only to put questions. There is no provision for the use

of the timing mechanisms necessary for administering the lexical decision task and for measuring the contestants' reaction times. One might as well allow the introduction of apparatus for measuring the contestants' magnetic fields or energy dissipation.

7.3.2. Rating Games[8]

The following are examples of questions that could be asked in the course of what French calls a 'rating game': on a scale of 0 (completely implausible) to 10 (completely plausible), rate ' "Flugbloggs" as a name Kellogg's would give to a new breakfast cereal', rate ' "Flugly" as the surname of a glamorous female movie star', 'rate banana splits as medicine' (ibid.: 18, 21).

French believes that questions like these will enable an interrogator to identify the computer. However, such questions may be of no assistance at all, since the computer is at liberty to attempt to pass itself off as a member of a foreign culture (e.g. as a tourist from rural Japan on his or her first trip overseas). Conveniently, French claims to discern 'an assumption tacit in Turing's article', namely that the computer must pass itself off as a member of the interrogator's own culture (ibid.: 15). French leaves it a mystery why Turing would have wished to impose a restriction which makes the test harder for the computer to pass and yet offers no conceptual gain. In fact, Turing says explicitly in the 1952 presentation that the computer is to 'be permitted all sorts of tricks so as to appear more man-like'. (Likewise the human foil need not be drawn from the same culture as the interrogator.)

French terms his rating game questions 'subcognitive', meaning that they probe the candidates' 'subconscious associative network ... that consists of highly overlapping activatable representations of experience' (ibid.: 16). This description of the questions is driven by connectionist theory, of course. An AI researcher might say with some justice that in so far as French's sample questions ('rate dry leaves as hiding places', 'rate pens as weapons', 'rate jackets as blankets', and so forth (ibid.: 20–21)) have any one thing in common, it is that the majority of them probe the candidates on their common-sense knowledge of the world. Viewed in this light, French's rating games fail, for the most part, to provide any new challenge. Nor can French assume that only connectionist devices will perform satisfactorily in these games: it remains to be seen how high a score can be obtained by a conventional computer equipped with a massive store of common sense knowledge, such as Lenat's presently incomplete CYC.

Turing himself envisaged that the process of constructing the machine that is to 'imitate an adult human mind' may involve subjecting a 'child machine' to 'an appropriate course of education', which would possibly include allowing the machine 'to roam the countryside' equipped with 'the best sense organs that money can buy' (1950a: 455, 456, 457, 460; 1948: 13). Turing canvasses the possibility that the 'child machine' should consist of an initially unorganised network of neuron-like elements (1948: 14–16). One might reasonably conjecture that the resulting adult

machine would do rather well in the Turing test. In the end, French's case against Turing's predictions of success in the imitation game rests on French's claim that unless a machine 'resembled us *precisely* in *all* physical respects', its experiences of the world would differ from ours in a way 'detectabl[e] by the Turing Test' (ibid.: 22, 23; my italics). However, French offers no argument whatsoever for this claim.

8. Prognosis

In 1953 Maurice Wilkes, head of the Cambridge University Mathematical Laboratory and designer of the EDSAC, wrote:

> If ever a machine is made to pass [Turing's] test it will be hailed as one of the crowning achievements of technical progress, and rightly so. (1953: 1231)

I agree entirely. A recent volume on Turing's work suggests that the Turing test be '[c]onsigned to history' (Millican and Clark, eds, 1996: 53). It will not be, of course. Perhaps Turing's modest prediction – a machine that is misidentified by 3 out of 10 average judges during five minutes of questioning – will indeed come to pass in the near future. As for Turing's predictions of more substantial success, one can only keep an open mind.[9]

Notes

* This paper formed the philosophy keynote address at *Turing 2000: The Dartmouth Conference on the Future of the Turing Test.*

1 The complete script is published for the first time in Copeland (1999).

2 Max Newman played an important part in Turing's intellectual life over many years. Newman brought Hilbert's ideas from Göttingen to Cambridge. In 1935, Turing attended lectures in which Newman discussed Hilbert's Entscheidungsproblem. In these lectures Newman introduced the concept that led Turing to his 'computing machines' (Turing, 1936): Newman defined a 'constructive process' as one that a *machine* can carry out (Newman in interview with Christopher Evans, 'The Pioneers of Computing: An Oral History of Computing', London: Science Museum (1976)). During the war Newman and Turing both worked at GC & CS, where the two cooperated closely. It was Newman who initiated the electronic decryption project that culminated in the construction of Colossus, the first large-scale electronic digital computing machine (designed by the engineer T.H. Flowers). At the end of the war Newman established the Royal Society Computing Machine Laboratory at Manchester University. Here he introduced the engineers F.C. Williams and T. Kilburn to Turing's idea of a universal computing machine, and under Newman's guidance Williams and Kilburn built the first stored-program electronic digital computer (see further Copeland 1999: 455–457). It was Newman who, in 1948, appointed Turing as Deputy Director (there being no Director) of the Manchester Computing Machine Laboratory.

3 Cottingham et al. (1985: 140).

4 The full text of the lecture is published for the first time in Copeland (1999).

5 Elsewhere Turing uses the verb 'simulate': 'My contention is that machines can be constructed which will simulate the behaviour of the human mind very closely' (Turing, 1951b).

6 See further Copeland (1996).

7 Page references are to the reprinting of French's article in Millican and Clark (1996).

[8] This subsection is indebted to Diane Proudfoot. See further Copeland and Proudfoot (1999b).

[9] Research on which this article draws was supported in part by University of Canterbury Research Grant no. U6271. Special thanks to Diane Proudfoot for her comments on a draft.

References

Block, N. (1981), ‘Psychologism and Behaviourism’, *Philosophical Review* 90, pp. 5–43.

Carpenter, B.E., Doran, R.W., eds, (1986), *A.M. Turing’s ACE Report of 1946 and Other Papers*, Cambridge, Mass.: MIT Press.

Copeland, B.J. (1993), *Artificial Intelligence: a Philosophical Introduction*, Oxford: Blackwell.

Copeland, B.J. (1996), ‘The Church-Turing Thesis’, in E. Zalta, ed., *The Stanford Encyclopaedia of Philosophy* [http://plato.stanford.edu].

Copeland, B.J. (ed.) (1999), ‘A Lecture and Two Radio Broadcasts by Alan Turing’, in K. Furukawa, D. Michie and S. Muggleton, eds, *Machine Intelligence* 15, Oxford University Press.

Copeland, B.J. (2001), ‘The Chinese Room from a Logical Point of View’, in J. Preston and M. Bishop, eds, *Views into the Chinese Room*, Oxford University Press.

Copeland, B.J. and Proudfoot, D. (1996), ‘On Alan Turing’s Anticipation of Connectionism’, *Synthese* 108, pp. 361–377.

Copeland, B.J. and Proudfoot, D. (1999a), ‘Alan Turing’s Forgotten Ideas in Computer Science’, *Scientific American* 280 (April), pp. 76–81.

Copeland, B.J. and Proudfoot, D. 1999b, ‘The Legacy of Alan Turing’, *Mind* 108, pp. 187–195.

Cottingham, J., Stoothoff, R. and Murdoch, D. (eds) (1985), *The Philosophical Writings of Descartes*, Vol. 1, Cambridge University Press.

de Cordemoy, G. (1668), *A Philosophicall Discourse Concerning Speech*, John Martin: London (repr. by Scholars’ Facsimiles & Reprints, New York, 1972).

French, R. (1990), ‘Subcognition and the Limits of the Turing Test’, *Mind* 99, pp. 53–65 (repr. in Millican and Clark 1996).

French, R. (2000), ‘The Turing Test: the First 50 Years’, *Trends in Cognitive Sciences* 4, pp. 115–122.

Heiser, J.F., Colby, K.M., Faught, W.S. and Parkison, R.C. (1980), ‘Can Psychiatrists Distinguish a Computer Simulation of Paranoia from the Real Thing?’, *Journal of Psychiatric Research* 15, pp. 149–162.

Hodges, A. (1992), *Alan Turing: The Enigma*, London: Vintage.

Millican, P. and Clark, A. (eds) (1996), *Machines and Thought: The Legacy of Alan Turing*, Oxford University Press.

Moor, J.H. (1976), ‘An Analysis of the Turing Test’, *Philosophical Studies* 30, pp. 249–257.

Moor, J.H. (1987), ‘Turing Test’, in S.C. Shapiro, ed., *Encyclopedia of Artificial Intelligence*, Vol. 2, New York: Wiley.

Narayanan, A. (1996), ‘The Intentional Stance and the Imitation Game’, in Millican and Clark (1996).

Osherson, D.N. and Lasnik, H. (eds) (1990), *An Invitation to Cognitive Science*, Vol. 3, Cambridge, Mass.: MIT Press.

Searle, J. (1980), ‘Minds, Brains, and Programs’, *Behavioural and Brain Sciences* 3, pp. 417–424.

Shannon, C.E. and McCarthy, J. (eds) (1956), *Automata Studies*, Princeton University Press.

Shieber, S.M. (1994), ‘Lessons from a Restricted Turing Test’, *Communications of the ACM* 37, pp. 70–78.

Turing, A.M. (1936), ‘On Computable Numbers, with an Application to the Entscheidungsproblem’, *Proceedings of the London Mathematical Society*, Series 2, 42 (1936–37), pp. 230–265.

Turing, A.M. (1945), ‘Proposal for Development in the Mathematics Division of an Automatic Computing Engine (ACE)’, in Carpenter and Doran (1986).

Turing, A.M. (1947), ‘Lecture to the London Mathematical Society on 20 February 1947’, in Carpenter and Doran (1986).
Turing, A.M. (1948), ‘Intelligent Machinery’, National Physical Laboratory Report, in B. Meltzer and D. Michie, eds, *Machine Intelligence 5*, Edinburgh University Press (1969).
Turing, A.M. (1950a), ‘Computing Machinery and Intelligence’, *Mind* 59, pp. 433–460.
Turing, A.M. (1950b), *Programmers’ Handbook for Manchester Electronic Computer*, Royal Society Computing Machine Laboratory, University of Manchester.
Turing, A.M. (1951a), ‘Can Digital Computers Think?’, in Copeland (1999).
Turing, A.M. (1951b), ‘Intelligent Machinery, A Heretical Theory’, in Copeland (1999).
Turing, A.M. (1952), ‘Can Automatic Calculating Machines Be Said To Think?, in Copeland (1999).
Von Neumann, J. (ed.) (1945), ‘First Draft of a Report on the EDVAC’, Moore School of Electrical Engineering, University of Pennsylvania.
Whitby, B. (1996), ‘The Turing Test: AI’s Biggest Blind Alley’, in Millican and Clark (1996).
Wilkes, M.V. (1953), ‘Can Machines Think?’, *Proceedings of the Institute of Radio Engineers* 41, pp. 1230–1234.